CSCI2244-Assignment 0
How to prepare the assignments
Howard Straubing

There is no Assignment 0, but let's pretend that there is one, and that it consists of performing the simulations in the file coin_experiments.py that we demonstrated in class, and answering a few questions based on the results of these simulations.

1. **What is the required file format and file name for the submissions?**

   Every submission should be in the form of a zipped folder. The folder itself should be named `FirstName_LastName_Assignment_x`, so for example if I were a student submitting this assignment, the folder would be named

   `Howard_Straubing_Assignment_0`

   and the submitted file

   `Howard_Straubing_Assignment_0.zip`

   Many students use a first name that is different from the one that appears on their official university records. But in this case, please use the official name---this makes it easier to assign grades if, as is often the case, several students have the same last name.

   The folder itself will contain at least the following two files:

   (1) A text-processed document containing thorough answers to the questions---what is meant by a thorough answer is illustrated in the examples below. **You must type your homework.** Typing mathematical formulas presents some difficulties. The simplest solution is to use MS Word and the Equation Editor, which is fairly self-explanatory and easy to use. That is what was used to prepare the present document, which was then converted to pdf. A better-looking solution is to use LaTeX, but this has a steeper learning curve. This document should be titled like the folder, so in this case, `Howard_Straubing_Assignment_0.pdf`.
   (2) A file containing any code you wrote for the assignment. Frequently, the assignment will specify exactly how you should name both this file and the functions it contains, as well as the parameter lists for these functions. You must use these prescribed names and formats---the hard-working TAs have to test your code, so a uniform format is really necessary.

   Many assignments will ask you to submit a plot produced with matplotlib. You can either include the image files generated by saving the plot in the folder itself, or better yet, incorporate them directly in the document, as is done here.

2. **If you make n tosses of a fair coin, does the number of heads approach n/2 as n grows larger? Support your answer with the results of a simulation.**

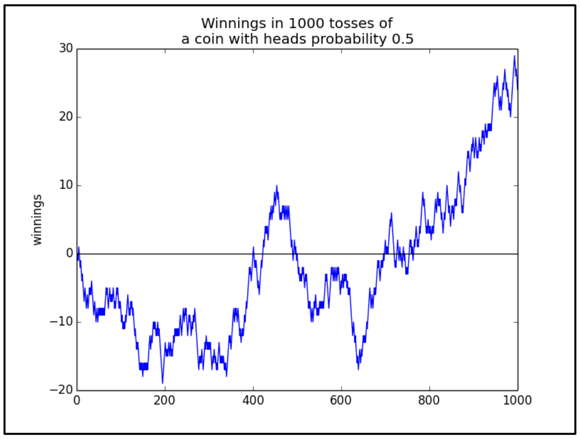
   See the function plot_hmt in the accompanying code. I plotted the difference

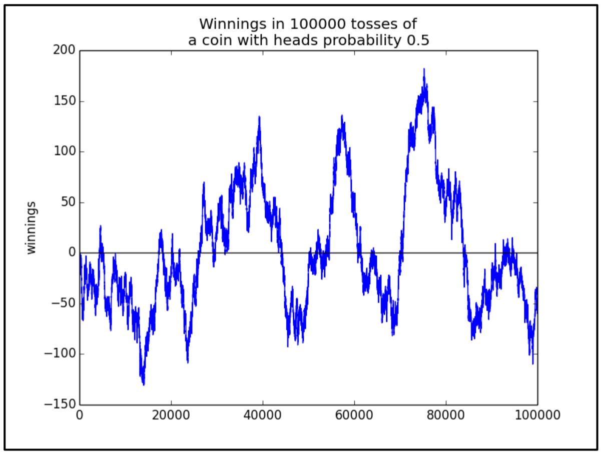
$$h_n - t_n$$

   where h_n, t_n are, respectively, the number of heads and tails in n tosses of the coin. Observe that

$$h_n - t_n = h_n - (n - h_n) = 2(h_n - n/2),$$

   so this is exactly two times the quantity in which we're interested. The plots below show several runs of the simulation for n=1000 and n=100,000



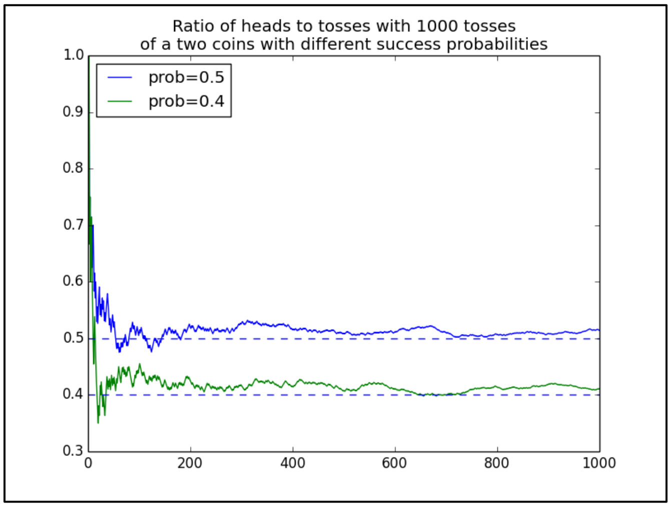


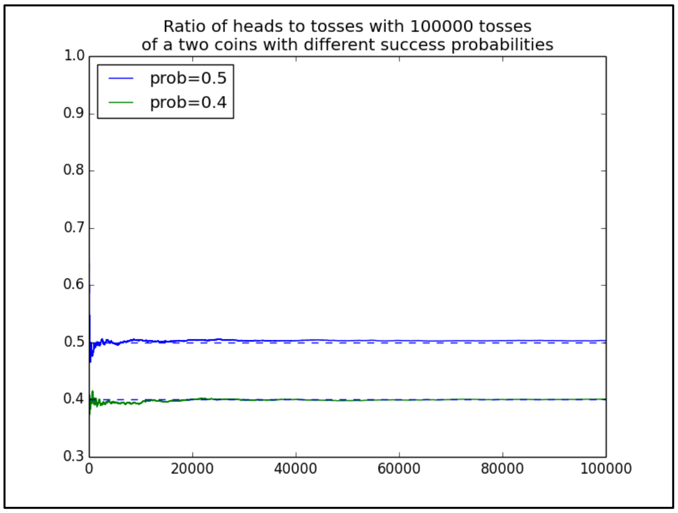
While there is quite a lot of fluctuation, the absolute deviation from n/2 actually *increases* as n gets larger (getting as large as 200 for 100,000 tosses but not more than 30 for 1000 tosses), instead of approaching 0. So the answer to the question is apparently No.

3. **What about the proportion of heads in the total number of tosses?**

   Another simulation in the attached code plots the ratio h_n/n, for two coins with different probabilities of heads. The plots below show this for n up to 1000, using both a fair coin (p=0.5) and an 'unfair' coin, with p=0.4, and for n up to 100,000. Here it really does appear that the proportion of heads approaches the probability of heads as n gets larger, in other words

$$\lim_{n \to \infty} h_n/n = p$$


Ratio of heads to tosses with 1000 tosses
of a two coins with different success probabilities
prob=0.5
prob=0.4


Ratio of heads to tosses with 100000 tosses
of a two coins with different success probabilities
prob=0.5
prob=0.4